

Mark scheme

Question			Answer/Indicative content	Marks	Guidance											
1	a		1 mark for each correct application software	5	Allow any sensible software. e.g. text editor for WP e.g. photo/image editor for row 1 BOD publication editor for DTP DNA brand names - ignore brand name if application type is also given. <u>Examiner's Comments</u> Many candidates were able to score at least 4 marks on this question. Those who gave brand names were not given marks as is standard for these types of question. Few candidates were able to give DTP for the fifth application type. Many gave email for the second application type which would not be the most appropriate for writing letters to clients.											
		<table><tr><th>Task</th><th>Application Software</th></tr><tr><td>Creating graphics such as a logo</td><td>Graphics/image editing software</td></tr><tr><td>Writing letters to clients to confirm their appointment date and time</td><td>Word processing software</td></tr><tr><td>Calculating the company profits at the end of each month</td><td>Spreadsheet/Accounting software</td></tr><tr><td>Storing, searching and updating client details and purchases</td><td>Database software</td></tr><tr><td>Creating brochures and flyers about the organisation</td><td>Desktop publishing software / DTP</td></tr></table>	Task			Application Software	Creating graphics such as a logo	Graphics/image editing software	Writing letters to clients to confirm their appointment date and time	Word processing software	Calculating the company profits at the end of each month	Spreadsheet/Accounting software	Storing, searching and updating client details and purchases	Database software	Creating brochures and flyers about the organisation	Desktop publishing software / DTP
		Task	Application Software													
		Creating graphics such as a logo	Graphics/image editing software													
		Writing letters to clients to confirm their appointment date and time	Word processing software													
		Calculating the company profits at the end of each month	Spreadsheet/Accounting software													
		Storing, searching and updating client details and purchases	Database software													
Creating brochures and flyers about the organisation	Desktop publishing software / DTP															
	b	i	1 mark per bullet to max 2: - Combines/links code/programs to files/software libraries... ...to form a single executable file	2	Allow one mark for valid description of static and/or dynamic linkers e.g. Static linkers combine code and libraries into one file / Dynamic linkers link/add addresses to libraries <u>Examiner's Comments</u> The majority of candidates were able to gain one mark on this question for stating that a linker links code with libraries but few went on to then describe it forming a single executable file.											
		ii	1 mark per bullet to max 2:	2												

			- It is part of the operating system - Loads an executable file (into memory)... ...from secondary storage - Loads the required software libraries		MP2 - Allow loads applications/programs <u>Examiner's Comments</u> Many candidates confused loaders with compilers and IDEs and talked about being able to use libraries in source code or loading code into the compiler.
			Total	9	
2			- Compiler Would protect their source code / keep their IP secret ... meaning competitors cannot steal their ideas ... Protect their money making ideas ...would mean the game would run quicker (than if it were interpreted).	AO2.1 (2)	<u>Examiner's Comments</u> Although many candidates correctly selected 'compiler' for the type of translator, this was often not followed up with a suitable reason.
			Total	2	
3	a	i	- Application software allows the user to solve a problem / complete a task. - Utility software is designed to monitor / manage / maintain a computer system	AO1.2 (2)	<u>Examiner's Comments</u> This question was generally answered well.
		ii	- Word Processor ... E.g. - Allows students to write essays / reports - Spreadsheets ... E.g. - Allow students to track/aggregate/calculate data - Create charts - Model situations / experiments - Presentation ... E.g.	AO1.1 (2) AO2.1 (2)	Allow any sensible application that could be used in a school environment as part of a student's workflow (e.g. photo editor, email client, web-browser). Example must be specific to a school environment Do not allow brand names for type of application If a brand name is used, allow follow through mark for a valid example <u>Examiner's Comments</u> Although this question was generally answered well, some candidates have used brand names for applications

			Allows students to create presentations on a particular topic		which is not allowed. Where a valid application was given, this was generally followed up with an appropriate example.												
		iii	- Open Source Software allows access/modification to/of the source code... ...can tailor the software to their specific needs ...may be more widely bug fixed ...there may be a wider pool of support available in the community Or - There will likely be no cost to purchasing a license to use the software... ...saving the school money	AO2.1 (2)	<u>Examiner's Comments</u> This question was generally answered well although some candidates stated that the software can be modified rather than stating that the source code can be modified.												
	b		e.g. - Disk defragmentation - File management - Device driver - System cleanup - Anti-virus/Anti-malware - Firewall - Backup - Compression	AO2.1 (3)	Allow other examples if valid (no duplicates) <u>Examiner's Comments</u> The majority of candidates have selected three utilities correctly in their response.												
			Total	11													
4			<table border="1"><thead><tr><th></th><th>Compiler</th><th>Interpreter</th><th>Assembler</th></tr></thead><tbody><tr><td>Creates an executable file</td><td>X</td><td></td><td>(X)</td></tr><tr><td>Creates one line of object code for each line of source code</td><td></td><td></td><td>X</td></tr></tbody></table>		Compiler	Interpreter	Assembler	Creates an executable file	X		(X)	Creates one line of object code for each line of source code			X	AO1.2 (4)	For “creates an executable file”, accept any of the following combinations: - Compiler - Assembler - Compiler and assembler <u>Examiner's Comments</u> Most candidates gained all four marks for this question. Although the question states to tick ‘one box’, candidates were given a mark if they
	Compiler	Interpreter	Assembler														
Creates an executable file	X		(X)														
Creates one line of object code for each line of source code			X														

			<table><tr><td>Translates all the high-level code at once</td><td>X</td><td></td><td></td></tr><tr><td>A program needs to be translated each time it is run</td><td></td><td>X</td><td></td></tr></table>	Translates all the high-level code at once	X			A program needs to be translated each time it is run		X			selected either assembler or compiler or both options for 'creates an executable file'.
Translates all the high-level code at once	X												
A program needs to be translated each time it is run		X											
			Total	4									
5		i	e.g. • Fewer mistakes (likely to be made) / More accurate • Faster as you can apply the same formula to multiple cells / By example • What-if analysis can be performed • Values can be changed and results automatically(re)calculated (by using formulas) • Can be shared electronically	1	Do not accept “faster” on its own without clarification of what/why it is faster. <u>Examiner’s Comments</u> Candidates who had a good knowledge of software were able to gain the mark on this question although many gave answers like faster or easier without saying what made it faster or easier.								
		ii	e.g. • Database/DBMS • ...to store/query/sort data about customers/staff/stock • Word processor • ...to create documents / letters / invoices for clients/staff • Presentation software • ...to create presentations for clients/staff • Email software • ...for staff to communicate with each other or with customers • Graphics manipulation • ...to produce adverts / images for sales • Web browser	4	Mark in pairs – one mark for naming type of application software, one for the example. Application type must be correct to give example. Do not accept brand names for first mark but FT for example. Ignore brand names if description given after E.g. Outlook / Email application Accept other sensible application software (such as CAD, Desktop Publishing). Do not accept special purpose / bespoke / utility software. Do not accept spreadsheet (given in question) Example must be relevant to the business								

			...to view websites to purchase materials/stock/ view competitor's website		<u>Examiner's Comments</u> Many candidates were able to identify application packages and could give valid tasks that a business could use them for. Less successful candidates tended to give utility software or were not able to apply the use to the business. Exemplar 1 Application 1: <u>Word processor</u> Example of task 1: <u>To write up letters to send to clients/employees.</u> Application 2: <u>Presentation Software</u> Example of task 2: <u>Could be used to pitch new ideas to the business.</u> The candidate has correctly identified 2 different applications and given a valid use for a business.
		iii	- No access to <u>source code</u> - Cannot modify/improve to meet business needs - Cannot fix bugs (Usually) cost to purchase licences / licence conditions to meet/ongoing fees	3	Do not award a reverse of the mark point by describing open source <u>Examiner's Comments</u> Candidates who understood that the source code is not available did well on this question as they were then able to expand their answer and gain full marks. Some candidates described open source without giving a drawback of closed source.
			Total	8	
6	a	i	To convert (high-level or assembly) code to <u>low level/machine code</u>	1	Do not allow answers referring to making the program executable, given in question. <u>Examiner's Comments</u> This question was generally well answered, though some candidates thought that a translator translated machine code into source code or that it translated code into something the computer could understand, without specifying what that was.
		ii	Compiler translates code all at once/before it's executed	4	Mark answers in pairs Max 2 marks per answer space

			• Interpreter translates code line by line / during runtime • Compiler produces executable file for reuse / Doesn't need to be translated everytime it is run • Interpreter needs to re-translate next time program is run • Compiler lists all errors/Compiled code doesn't run if there are any errors • Interpreter stops at the first error • Compiled programs have the source code hidden • Interpreted programs have the source code visible		<u>Examiner's Comments</u> This question was well answered by many candidates who showed two distinct differences.																							
	b	<table><tr><th></th><th>Lexical analysis</th><th>Syntax analysis</th><th>Code generation</th></tr><tr><td>Comments and white-space are removed</td><td>x</td><td></td><td></td></tr><tr><td>Keywords are replaced with tokens</td><td>x</td><td></td><td></td></tr><tr><td>Object code is created</td><td></td><td></td><td>x</td></tr><tr><td>Symbol table created for variables</td><td>x</td><td></td><td></td></tr><tr><td>Builds an abstract syntax tree</td><td></td><td>x</td><td></td></tr></table>		Lexical analysis	Syntax analysis	Code generation	Comments and white-space are removed	x			Keywords are replaced with tokens	x			Object code is created			x	Symbol table created for variables	x			Builds an abstract syntax tree		x		5	One mark per row. No mark if more than one/no box is ticked. Accept other marks that clearly indicate choice (e.g. X)
	Lexical analysis	Syntax analysis	Code generation																									
Comments and white-space are removed	x																											
Keywords are replaced with tokens	x																											
Object code is created			x																									
Symbol table created for variables	x																											
Builds an abstract syntax tree		x																										

	c		- To make the program run faster/ code is more efficient - To make the program use fewer resources/less memory	2	<u>Examiner's Comments</u> Many candidates gave good descriptions of how code is optimised, but they did not answer the question which asked what the purpose of optimisation is.
			Total	12	
7	a		1 mark per bullet up to a maximum of 2 marks, e.g: - Word Processing - Writing letters to customers - Spreadsheets - Completing accounts - Presentation Software - Create\Show business plans - DTP - Creating marketing literature - Graphics Package - Editing photographs of procedures/marketing photos	2 (AO1.1) (1) (AO2.1) (1)	1 Mark for a suitable package and 1 mark for a relevant example for that package. Do not allow: - Non-business software (E.g. games) - Brand names (e.g Word/ Excel) - Database software <u>Examiner's Comments</u> This question was generally answered well. Several candidates wrote brand names which was not an acceptable answer. Some candidates gave an example of a utility which was required for Question 2 (b). To gain both marks, candidates were required to choose a suitable application type which would be useful in the given scenario. Candidates are reminded not to use brand names for questions of this type.
	b		1 mark per bullet up to a maximum of 2 marks, e.g: - Disk Defragmentation... ...To keep optimal r/w speed for her HDD - File management... ...To allow easy access to her file system - Disk Drivers... ... To allow her to use new peripheral devices - System Clean-up...	4 (AO1.1) (2) (AO1.2) (2)	1 Mark for a suitable utility and 1 mark for a relevant example for that utility. Do not accept task manager Accept: - Compression Software to make the file size smaller - Backup Software to make copies of files

		... to keep her system free of redundant files - Anti-Virus/Malware... ... to find and remove/quarantine viruses/malware (A worms/trojans etc)		<u>Examiner's Comments</u> This question was generally answered well although many candidates achieved 2 rather than 4 marks. Commonly, two utilities were correctly identified but the explanation to accompany the utility was too vague. Candidates were required to give examples of how the utilities could be used with the given scenario.
	c	Mark Band 3–High Level (7-9 marks) The candidate demonstrates a thorough knowledge and understanding of open and closed source software; the material is generally accurate and detailed. The candidate has covered all 4 sections of cost, usability, security and Support available and for the top of this mark band will have covered all 4 well. Evidence/examples will be explicitly relevant to the explanation. <i>There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.</i> Mark Band 2-Mid Level (4-6 marks) The candidate demonstrates reasonable knowledge and understanding of open and closed source software; the material is generally accurate but at times underdeveloped. The candidate has covered all at least 2 of the 4 sections of cost, usability, security and Support available. Evidence/examples are for the most part implicitly relevant to the explanation. <i>There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.</i> Mark Band 1-Low Level (1-3 marks)	9 (AO1.1) (2) (AO1.2) (2) (AO2.1) (2) (AO3.3) (3)	<u>Knowledge:</u> Cost Open Source (generally) free to use - May have to purchase maintenance contracts - Staff training if “non-standard” Closed Source (sometimes) have to pay to license the software - If paid will (usually) it will come with some level of support Usability/extensibility Open Source - Tends to have a lower focus on UI - Source code released (under license) - Source can be edited - Can be redistributed (under license) Closed Source - Professionally developed - Distributed with a restrictive license - Only executable/object code is distributed/source code not distributed - Cannot be redistributed Security Open Source

			The candidate demonstrates a basic knowledge of how the layers of open and closed source software; the material is basic and contains some inaccuracies. The candidate makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides nothing more than an unsupported assertion. <i>The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.</i> 0 marks No attempt to answer the question or response is not worthy of credit.		• potentially massive bank of volunteer developers working on the product • Many of the contributors may not be professional • Code available to be scrutinised by anyone... • ...but this may include people with malicious intentions Closed Source • Closed teams of developers • More work scrutiny for code • security fixes usually addressed quicker Support Available Open Source • Source code released (under license) • Source can be edited • Open communities mean lots of support options could be available <u>Closed Source</u> • <u>Support may be available from the company producing the software.</u> <u>Application:</u> Cost Open Source • Lower overheads to company • Extra staff training and hardware cost could lead to total cost of ownership being higher Closed Source • Support from vendor can lead to quicker fixes. Usability Open Source
--	--	--	---	--	--

					• The ability to edit source code means bespoke functionality can be developed in house • Lower focus on UI can mean a harder to use product (leading to higher training costs) Closed Source • Due to professional development, finish tends to be a higher standard • As the organisation protects their IP they generally tend to be less buggy as the organisation reputation/business model will rely on it • Lack of source code means extra features can only be developed by the vendor. Security Open Source • Tends to be less secure as more people working on it, not always under rigorous oversight • No paid developers mean people may not work on security fixes straight away Closed Source • Developers work under tighter standards • Code being scrutinised more will mean less likely to be ship with bugs • Professional standards lead to quicker turnaround of bugs Support Available Open Source • Editable source code means could self-support • Open communities mean there is vast amounts of knowledge available.
--	--	--	--	--	---

					<u>Evaluation:</u> Open Source - Open source would lead to potential cost savings if Charlie looked to self support by using online communities or handling the code herself. Closed Source - If Charlie lacked technical skills, the better UI design from professional developers may make the UX smoother - Charlies would have a legal obligation to the data stored for her business. Closed source tighter security may strengthen this - Charlies business will rely on the uptime of her system. Professional support offered by the developers may mean less downtime in the case of software issues. <u>Examiner's Comments</u> Candidates were assessed on the quality of their extended response in this question. Most of the candidates were clear on the basic difference between open and closed source software but did not link this in with the scenario. Mid-level responses were common with few candidates having a balanced discussion with coverage of all four of the required sections.
			Total	15	
8	a		Assembler	1 AO1.1 (1)	
	b	i	Iteration	1 AO2.2 (1)	
		ii	5	1 AO2.2 (1)	

		iii	1 mark per bullet up to a maximum of 4 marks, e.g: - Initialise Y and Z AND set X - Correct use of IF - Correct condition (e.g. if X >= Y then) - Assignment of Z in correct places	4 AO3.1 (2) AO3.3 (2)	X Y Z variable alternatives are acceptable Solution: <pre>X = input() Y = 5 Z = 0 if X >= Y then Z = Y else Z = X Endif</pre>
			Total	7	
9	a	i	- Source code / program code is freely available ...to edit/amend recompile.	2 AO1.1	
		ii	e.g. - Can modify code and adapt IDE to her needs - Is likely to be financially free of cost. - Can recompile to work on different systems - Has the benefit of a community potentially improving the system - Can learn from others - Can ensure no backdoors / malware	1 AO2.1	Do not accept simply seeing the code (previous question).
	b	i	- Sections of code / program file - Written by other authors / already written - Containing useful routines - Suitable example (e.g. GUI routines, database access routine, encryption, graphics)	3 AO1.1 (2) AO2.1 (1)	Maximum 2 for definition, 1 for example
		ii	- Save time... ...because no need to rewrite code - Use expertise of others.. ...to complete tasks that require specialist knowledge / abstract away complexity. - Has already been tested/the programmer doesn't have to test it themselves	2 AO1.2	

			• Making debugging easier/saving time		
		iii	• May (significantly) increase size of compiled file... • ...as library contains many routines that aren't being used. • Not written by the programmer • ...so introduces uncertainty / require further testing / programmer needs to spend time familiarising themselves with it	2 AO1.2	
		iv	• Links the main program to libraries • ..can either include them in the final executable code • ..or get the executable code to point to the external libraries	3 AO1.2	Accept terms static and dynamic for bullets 2 and 3, but only if these are explained.
			Total	13	